

Avis d'expert, 25 octobre 2011

## **Activités de test externalisées : 5 conseils pour augmenter les chances de succès**

Le sujet de la « sous-traitance d'activités de test » est présent dans tous les projets de développement informatique, ou presque, mais c'est souvent la question de l'Off-shore ou du Near-Shore qui l'anime. Certes, la localisation géographique des activités de test a une importance qui n'est pas des moindres, vue les spécificités qu'elle peut introduire, mais il peut être aussi très utile de penser à mettre en place des outils ou méthodes simples et efficaces pour augmenter les chances de succès de la sous-traitance dans le domaine du test logiciel.

**Par Eric RIOU du COSQUER, Trésorier du CFTL (Comité Français des Tests Logiciels), Secrétaire de l'ISTQB et TestManager Senior chez France Télécom.**

Diplômé de l'Ecole d'Ingénieurs de Brest (Informatique Industrielle), Eric Riou du Cosquer a commencé sa carrière en 2000, en qualité de Chef de Projet au sein des sociétés CSC PeatMarwick, puis Softeam de 2000 à 2003, avant de rejoindre la division R&D de France Télécom. Depuis 2007, Eric exerce en qualité de Consultant et Formateur l'ensemble du groupe France Télécom, en France et à l'étranger. Il participe également à l'amélioration continue des processus et pratiques de test et est amené à conduire des audits internes ou externes pour différentes divisions de France Télécom.

Depuis 2002, il a dispensé plusieurs conférences et formations dans le domaine des tests de logiciels. En tant qu'expert, Eric Riou du Cosquer est également membre du groupe de travail international pour la définition du Syllabus Avancé de l'ISTQB, Responsable de la création de la version française et fût Président du Club utilisateurs de Mercury France, de 2008 à 2011.

### **De quoi parle-t-on ?**

Dans le cadre d'un projet informatique, le chef de projet fait en général appel à des personnes n'appartenant pas à l'équipe du projet.

Ces « Personnes n'appartenant pas à l'équipe du projet » peuvent appartenir à la même société ou bien à des sociétés externes, travaillant comme sous-traitants de la société porteuse du projet.

Elles peuvent appartenir à une entité spécialisée dans un domaine précis ou au contraire être des ressources indépendantes mises à la disposition des projets. Enfin, ces personnes peuvent être localisées dans le même pays que le chef de projet ou dans un pays plus ou moins éloigné.

Dans cet article, le mot « externalisé » désignera ce qui n'est pas fait par des personnes intégrées à l'équipe du projet.

Ceci est valable pour une grande partie des activités liées au développement logiciel, y compris pour les principales activités de test qui peuvent se décliner sur différents niveaux de test. Pour rappel, le syllabus niveau fondation de l'ISTQB propose 5 ensembles d'activités :

E Planifier et contrôler

E Analyser et concevoir

E Implémenter et exécuter

E Évaluer les critères de sortie et informer

E Activités de clôture des tests

Ces activités peuvent se décliner sur 4 niveaux de test :

E Composants

E Intégration

E Système

E Acceptation

La question qui se pose maintenant est de savoir si oui ou non tout peut être sous-traité. Théoriquement, oui. Dans la pratique, certaines tendances se dégagent :

E les activités de test liées au niveau composant sont souvent portées par l'entité en charge des développements correspondants, et se déroulent avant, pendant et après le développement pur et dur. En d'autres termes, si le développement d'un composant est sous-traité, alors les activités de test associées, ont de fortes chances de l'être aussi.

E les activités de test liées aux niveaux Intégration et Système sont sous-traitées environ une fois sur deux.

E les activités de test liées au niveau Acceptation ne sont en général pas sous-traitées, compte tenu des compétences métiers nécessaires, ainsi que de l'utilisation de données confidentielles et d'environnements techniques complexes et coûteux.

Pour les projets suivant une démarche Agile, la situation est différente mais, c'est un autre sujet !

Dans cet article, la notion de « sous-traitance d'activités de test » désignera essentiellement, les activités appliquées au niveau composant lorsque le développement est sous-traité et les activités mises en œuvre aux niveaux Intégration et système.

### **Le cas d'une grande entreprise internationale qui lance un grand projet de transformation**

Une grande société, présente dans différents pays, fait évoluer son Système d'Information (SI) dans le cadre d'un vaste projet de transformation. Le SI est constitué d'un ensemble d'applications couvrant différents domaines techniques ou métiers. Ce projet va donner lieu à l'introduction de nouvelles applications et à la modification ou suppression de certaines autres applications selon les exigences définies par la société porteuse du projet global.

Différents sous-traitants se partagent le développement des applications, un sous-traitant étant en charge de plusieurs applications.

Un sous-traitant, différent des précédents, est chargé de l'ensemble des tests d'intégration.

Enfin, la société porteuse du projet global est aussi chargée des tests d'acceptation.

### **Conseil n°1 : Mesurer les compétences de chaque sous-traitant et en tirer les conséquences.**

Ceci doit être fait au niveau de l'entreprise, mais aussi et surtout au niveau des personnes impliquées dans le projet, et ce de manière constructive et factuelle.

Une restitution doit être partagée avec le sous-traitant et déboucher, si nécessaire, sur un plan d'amélioration suivi de près.

Il est en général utile d'avoir dans sa boîte à outils des connaissances sur :

E CMMI

E TMMI

E ISTQB/CFTL (Différentes normes et standards référencés : IEEE 829, IEEE 1044, ISO 9126...)

Cependant, avant de sortir l'artillerie lourde, il est en général prudent de vérifier que le sous-traitant maîtrise les bases incontournables du test logiciel. Pour cela, un simple questionnaire peut suffire à déceler de vraies lacunes. Ainsi, par exemple, un sous-traitant qui, à la question « Quel outil utilisez-vous pour gérer les défauts ? » répond, comme cela m'est arrivé récemment, « Nous utilisons le mail et le téléphone » aura à priori un niveau très faible en test logiciel. Il conviendra alors de définir un plan d'amélioration pour le sous-traitant et de le suivre pour assurer la mise en place de pratiques incontournables, comme celles indiquées dans l'exemple ci-dessous

Attention, il n'est pas acceptable d'entendre un sous-traitant en charge du développement d'un certain nombre de composants dire « Je suis responsable des développements, vous serez livré mais je n'ai aucune preuve à vous donner sur les tests que je réalise »...

Si cela se produit, la séquence de questions suivante doit permettre d'aider le sous-traitant à rapidement retrouver le bon sens qu'il avait perdu :

E « Allez-vous me livrer un composant sans l'avoir testé ? »

E « Avez-vous réfléchi à la façon de le tester ? »

E « Avez-vous documenté votre stratégie de test ? »

E « Allez-vous exécuter des tests ? »

E « Où ces tests sont-ils décrits ? »

E « Quels sont les résultats de ces tests ? »

Bien sûr, toute réponse doit s'accompagner d'un document faisant office de preuve !

### **Conseil n°2 : Identifier et définir le plus haut niveau de test**

Quand une partie des activités de développement et de test est externalisée, il devient plus difficile d'avoir une vision globale sur l'ensemble des activités de test d'un projet. Pourtant c'est indispensable pour bien gérer le projet de test et coordonner les activités qui vont se dérouler aux différents niveaux.

La notion de « Plan de test maître » (« Master Test Plan ») telle qu'introduite par la norme IEEE 829-2008 prend alors tout son sens. Ce plan de test maître permettra, non seulement de définir et contrôler une planification globale de l'activité de test sur l'ensemble du projet, mais aussi d'identifier les différents niveaux de test et les Plan de Test associés.

Pour un composant dont le développement est sous-traité, il conviendra alors de demander au sous-traitant, s'il est également en charge des tests, un Plan de Test de niveau composant. Un Plan de Test de niveau Intégration pourra être demandé à un sous-traitant en charge de l'intégration. Externaliser une activité de test ne veut pas dire « perdre contrôle et visibilité » sur celle-ci.

Dans l'exemple ci-dessous, un « Master Test Plan » a permis de définir une stratégie globale de test pour un projet complexe d'évolution d'un SI, faisant appel à de nombreux sous-traitants. Par exemple, chaque composant développé a donné lieu à la rédaction d'un Plan de Test référencé par le Plan de Test maître.

### **Conseil n°3 : Généraliser les revues qui sont un outil d'une rentabilité exceptionnelle !**

Qu'est ce qu'une revue ? C'est tout simplement une « évaluation d'un état d'un produit ou projet pour déceler des déviations par rapport aux résultats planifiés et recommander des améliorations » [IEEE 1028]

Qu'est ce qui peut être revu ? Tout ou presque ! Cependant, lorsque des activités de développement et de test sont externalisées, les revues présentent un intérêt particulier pour 3 produits qui sont rarement revus :

E les plans de Test

E les tests

E les bordereaux de livraison

Même si c'est un sous-traitant qui est en charge des tests unitaires du composant qu'il développe, revoir le Plan de Test associé permettra d'avoir des garanties sur la façon avec laquelle ce composant sera testé. L'exemple ci-dessous montre le résultat d'une revue faite sur 5 plans de test réalisés par un sous-traitant sur la base d'un même template. Chaque ligne correspond à un plan de test et chaque colonne à un chapitre du plan de test. Cette revue a mis en évidence une grande faiblesse concernant le test chez le sous-traitant et a donné lieu immédiatement à un plan d'action d'amélioration.

Les tests eux-mêmes peuvent aussi être revus. Il faudra bien sûr vérifier que chaque test est associé à une caractéristique à tester (ou exigence), que le format est correct, mais aussi que le test est pertinent. La revue des tests doit être effectuée par des personnes ayant les bonnes compétences, techniques ou fonctionnelles.

Enfin, la revue des bordereaux de livraison peut également apporter beaucoup. Ce document est en effet primordial pour assurer le démarrage de l'intégration dans de bonnes conditions. Un composant livré avec un bordereau de livraison incomplet, sans procédure d'installation par exemple, posera des problèmes à l'intégration. L'idéal est même d'organiser des revues anticipées des bordereaux de livraison. Rien n'empêche en effet de demander à un sous-traitant de fournir une première version de chaque bordereau de livraison une semaine avant la date de

livraison finale. Cela permet aux personnes en charge de l'intégration de les revoir, de vérifier leur qualité et de demander les corrections nécessaires, sans retarder la livraison !

#### **Conseil n°4 : L'analyse de code outillée : un outil à utiliser absolument, mais avec maîtrise !**

L'analyse statique de code, à la différence des revues de code classiques faites par des développeurs, est en général réalisée avec un outil de test spécifique. Le principe est simple : le code développé va être analysé, sans être exécuté, afin d'évaluer sa qualité selon différents critères, bonnes pratiques ou règles de codage.

Lorsque les développements sont sous-traités, l'utilisation de ce type d'outil est très intéressante car elle peut aider le sous-traitant à respecter les règles de développement et les exigences sur la qualité formulées par l'entreprise client. L'outil peut également permettre de vérifier la qualité d'un code livré avant de l'accepter.

Toutefois, cet outil doit être bien maîtrisé, car différentes difficultés - nous en verrons deux ci-après - peuvent se présenter. Heureusement à chaque difficulté correspond en général une solution !

**Première difficulté** : si l'utilisation d'un tel outil, avec des critères d'acceptation de livraison, n'a pas été contractualisée avec le sous-traitant, il risque de refuser de modifier son code. L'idéal est bien sûr d'avoir défini les règles du jeu au départ et de permettre au sous-traitant d'analyser son code au fil du développement. Mais si cela n'a pas été fait, ce n'est pas une raison pour se priver d'analyser le code livré. Il faudra alors considérer 2 types de violations de règles de codage :

E des règles basiques, incontestables, qui correspondent à l'état de l'art en matière de développement et qu'il n'est pas possible de contourner, au risque de mettre en péril le système dans lequel le code sera intégré. Sur ces règles, il est légitime d'être intransigeant et d'exiger les modifications nécessaires de la part du sous-traitant, sans supplément de coût.

E des règles propres à l'entreprise et à ses objectifs spécifiques en termes de qualité de code : si ces règles n'ont pas été précisées à l'avance, il sera sans doute nécessaire d'établir un avenant et d'accorder des jours supplémentaires au sous-traitant pour les appliquer.

**Deuxième difficulté** : le sous-traitant est chargé de faire évoluer un code existant pour ajouter, supprimer ou modifier des fonctionnalités. Il faudra bien sûr veiller à faire la distinction entre le code d'origine et le nouveau code, mais cela ne suffit pas. En effet, faire évoluer un code, qui ne respecte pas des règles de codage, contraint souvent à rester dans le non respect des règles. Par exemple, faire évoluer une fonction, ayant un nombre trop important de lignes de code ou un nombre insuffisant de commentaires, ne permettra pas de respecter les règles relatives à la taille d'une fonction et aux commentaires attendus. Dans ce cas il est conseillé de coopérer intelligemment avec le sous-traitant, d'une part pour fixer des objectifs réalistes sur une partie de ses développements, et d'autre part pour envisager la réécriture d'une partie du code. Bien sûr cela a un prix, comme tous les aspects contribuant à l'amélioration de la qualité !

Un outil comme CAST donne une note globale sur la qualité du code analysé, mais également plusieurs vues aussi détaillées que souhaité, pour voir précisément quelles règles ne sont pas respectées et à quel endroit !

#### **Conseil n°5 : Construire une relation de confiance avec le sous-traitant**

Enfin, cela va de soit, mais il est important de le rappeler : le moyen le plus efficace de travailler avec des sous-traitants, dans le domaine du test mais aussi pour l'ensemble des activités du développement logiciel, est de bâtir

une relation de confiance intelligente. Des relations figées sur un contrat défini au début du projet avec des délais, des coûts et des critères d'acceptation inflexibles, auront peu de chance de se transformer en succès mutuel.

Il est important de contrôler l'activité de test des sous-traitants, mais il est important aussi de leur préciser des attentes et de les aider à les respecter. Être ferme mais coopérant !

Cela peut, et même doit, se traduire par des échanges réguliers qui prendront la forme de réunions de suivi entre tous les acteurs, mais aussi de visites dans les locaux du sous-traitant. Établir un contact et dialoguer de temps en temps avec les équipes du sous-traitant pourront montrer aux personnes impliquées, manager ou « petites mains », qu'ils travaillent pour un client qui a un visage et qui compte sur la qualité de leur travail.

Lorsque plusieurs sous-traitants sont impliqués dans un grand projet, il est aussi très utile de les rassembler pour arbitrer les choix et gérer au mieux les difficultés et conflits éventuels entre sous-traitants.

Vous l'aurez compris, au-delà des points de vigilance classiques mis en œuvre lorsque certaines activités de test sont externalisées, il existe des moyens simples et efficaces à appliquer pour augmenter les chances de succès de l'externalisation.

#### **Eric RIOU du COSQUER**

Trésorier du CFTL (Comité Français des Tests Logiciels)

Secrétaire de l'ISTQB

Test Manager Senior chez France Télécom